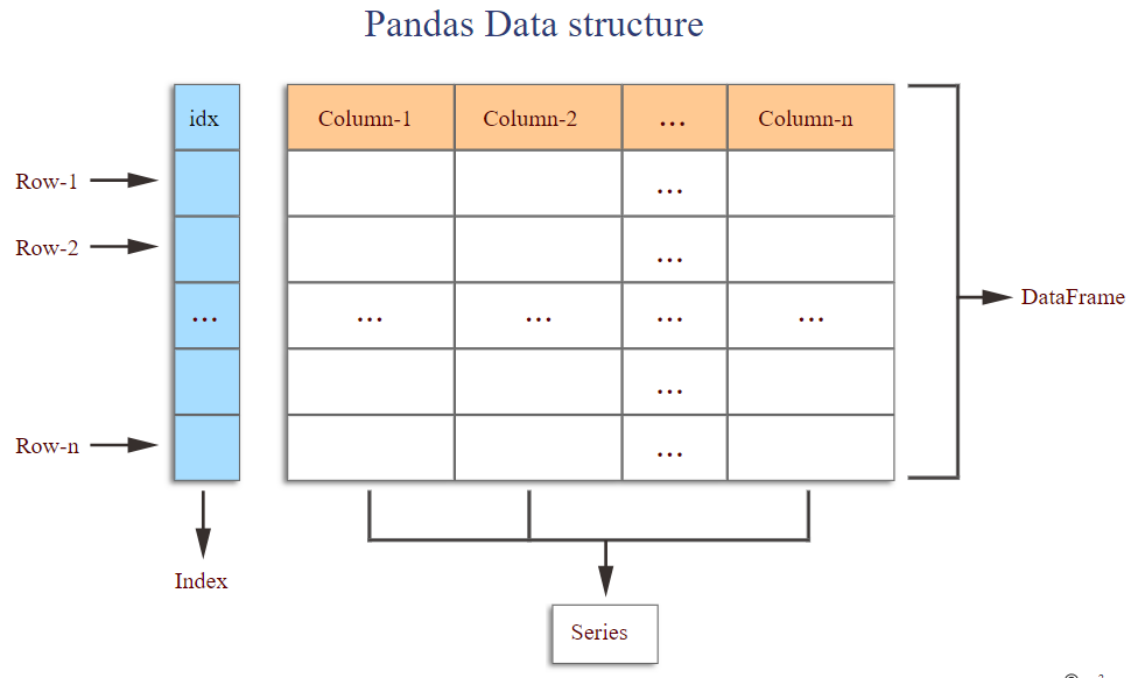
PYTHON APLICADO A LA CLIMATOLOGÍA

Bch. Javier Chiong Ravina

Pandas

- Es una librería que permite procesar datos tabulares (o tabla de datos) como hojas de cálculo o base de datos.
- En pandas, a las tabla de datos se les llama Dataframe. Las dataframe se caracterizan por estar compuestas de filas (rows) y columnas (columns). Al nombre que describe cada columna se le llama column y al nombre que describe cada fila se le llama index.



Atributos de un DataFrame

Existen varias propiedades o métodos para ver las características de un DataFrame.

df.info() : Devuelve información (número de filas, número de columnas, índices, tipo de las columnas y memoria usado) sobre el DataFrame df.

df.shape : Devuelve una tupla con el número de filas y columnas del DataFrame df.

df.size : Devuelve el número de elementos del DataFrame.

df.columns : Devuelve una lista con los nombres de las columnas del DataFrame df.

df.index : Devuelve una lista con los nombres de las filas del DataFrame df.

df.dtypes : Devuelve una serie con los tipos de datos de las columnas del DataFrame df.

df.head(n) : Devuelve las n primeras filas del DataFrame df.

df.tail(n) : Devuelve las n últimas filas del DataFrame df.

Pandas

- La importación de la librería se ejecuta con el siguiente comando:

import pandas as pd

Funciones de la librería Pandas mas usadas:

- pd.date_range() : Se utiliza para crear un rango de fechas
- pd.Series() : Se utiliza para dar la estructura de dataframe a una serie de datos de 1D.
- pd.Timestamp() : Se utiliza para crear fechas y horas
- pd.Categorical() : Se utiliza crear una lista de parámetros, el cual contará con algunos métodos.
- pd.DataFrame() : Se utiliza para crear un Dataframe

Importación de ficheros

Para importar archivos con formato txt, csv y Excel se pueden realizar las siguientes instrucciones.

- pd.read_csv(fichero.csv, sep=separador, columns=booleano, index=booleano) : Importa el DataFrame df a partir de un documento con formato csv o txt. Sep indica el separador de caracteres que va a considerar la función para dividir los caracteres del documento en formato csv o txt.

- pd.read_excel(fichero.xlsx, sheet_name = hoja, columns=booleano, index=booleano) : Importa el DataFrame df a partir de un documento en formato Excel. Si se le asigna True al parámetro columns se importa también la fila con los nombres de columnas y si se le asigna True al parámetro index se importa también la columna con los nombres de las filas.

Funciones aplicadas a DataFrames

df.count() : Determina el número de elementos que no son nulos ni NaN en la serie s.

df.sum() : Determina la suma de los datos del dataframe o serie cuando los datos son de un tipo numérico, o la concatenación cuando son del tipo cadena str.

df.cumsum() : Devuelve un dataframe o serie con la suma acumulada de los datos de la serie s cuando los datos son de un tipo numérico.

df.value_counts() : Devuelve un dataframe o serie con la frecuencia (número de repeticiones) de cada valor del dataframe o serie.

df.min() : Devuelve el menor de los datos del dataframe o serie .

df.max() : Devuelve el mayor de los datos del dataframe o serie .

df.mean() : Devuelve la media de los datos del dataframe o serie cuando los datos son de un tipo numérico.

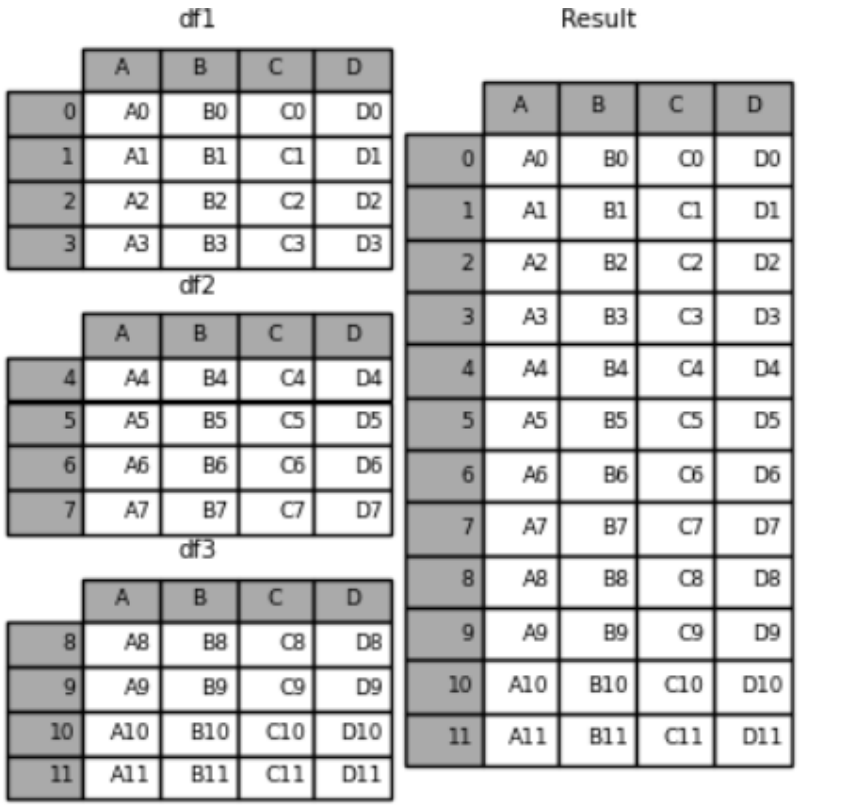
df.std() : Devuelve la desviación típica de los datos del dataframe o serie cuando los datos son de un tipo numérico.

df.describe(): Devuelve un dataframe o serie con un resumen descriptivo que incluye el número de datos, su suma, el mínimo, el máximo, la media, la desviación típica y los cuartiles.

Concatenación usando append

Para concatenar dataframes se usa el comando:

- df.append(other, ignore_index=False, sort=False) : Une los dataframes en base al nombre de las columnas.



	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

Fuente: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html

left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2

Fuente: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html

Unir columnas de un dataframe a otro

Para enlazar columnas de un DataFrame a otro se puede usar el comando merge.

- ***df.merge(right, how='inner', on=None)***

“How” nos va a indicar que dataframe usar de referencia y el comando “on” nos va a indicar en base a que columna vamos a unir los dataframes o columnas.

Reindexar un DataFrame

Change Column Order in Pandas

Country	year	gdp
A	1990	5
B	1990	2
A	2000	40
B	...	..

year	gdp	country
1990	5	A
1990	2	B
2000	40	A
...	...	..

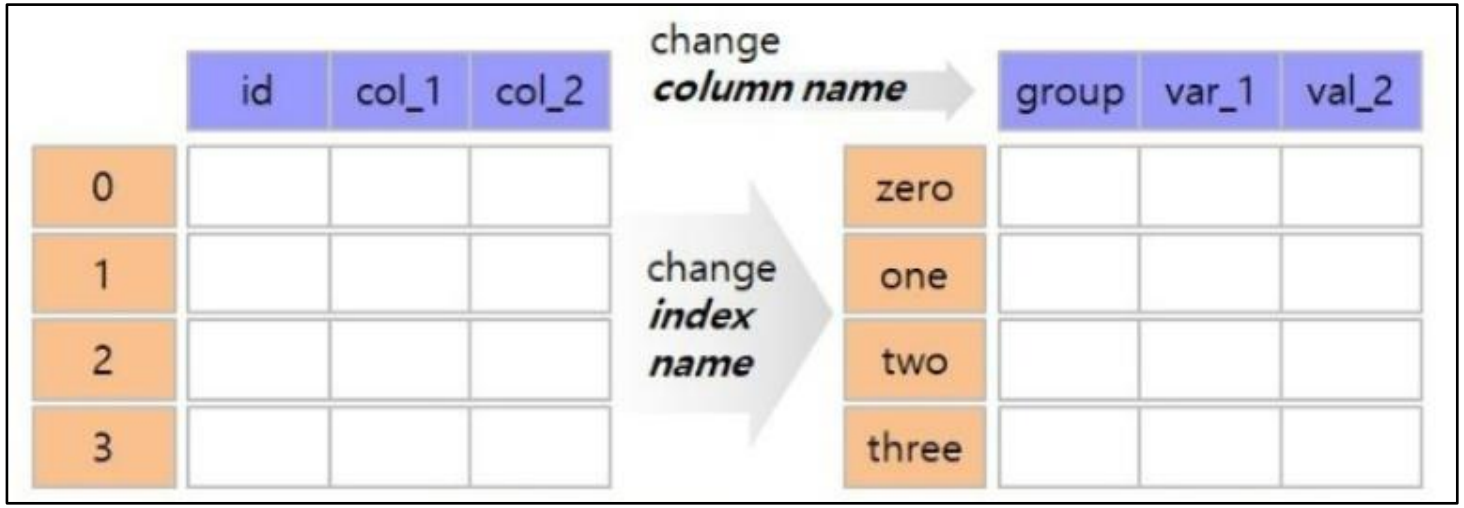
- Para reordenar los índices de las filas y las columnas de un DataFrame, así como añadir o eliminar índices, se utiliza el siguiente método:

- df.reindex(index=filas, columns=columnas, fill_value=relleno)

Renombrar los nombres de las filas y columnas

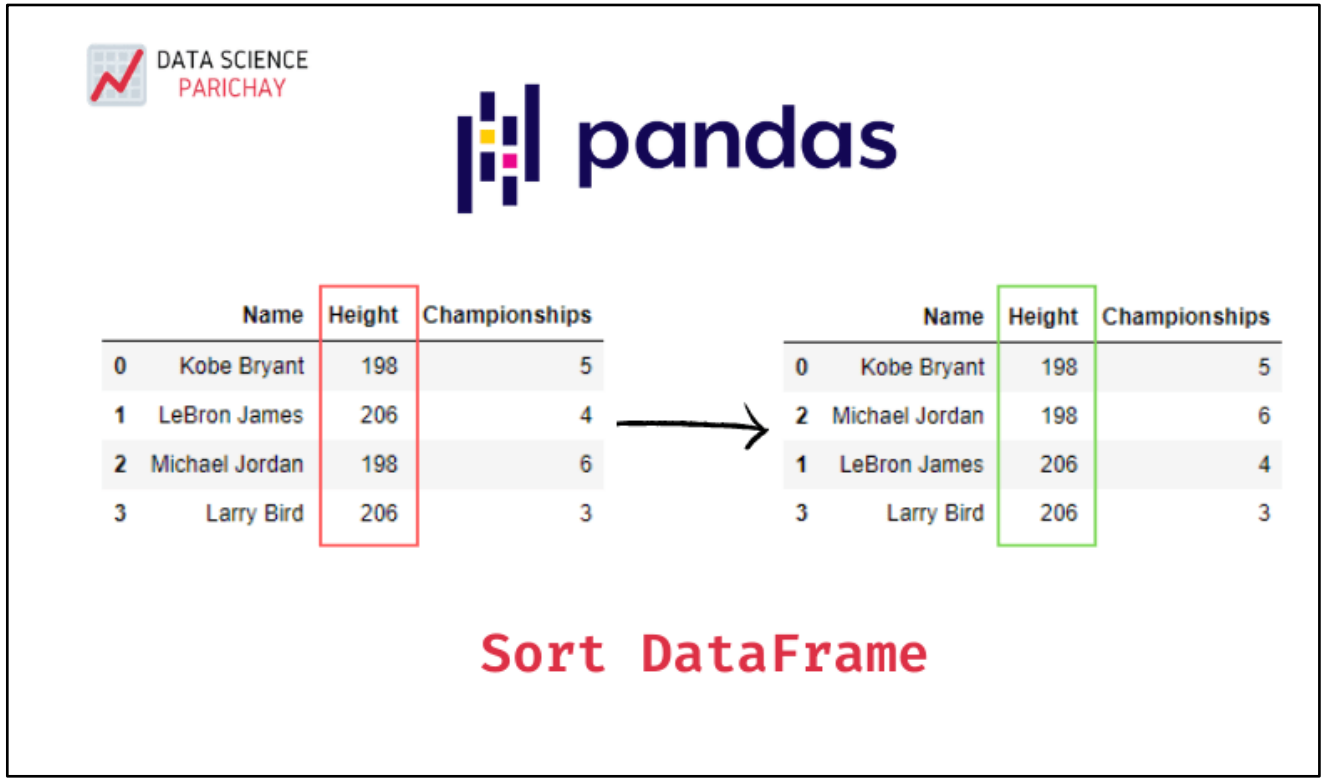
Para cambiar el nombre de las filas y las columnas de un DataFrame se utiliza el siguiente método:

- df.rename(columns=columnas, index=filas): Renombrar las columnas indicadas. En el parámetro columns se detalla las columnas que se quieren renombrar y el nuevo nombre de las columnas a través de un Diccionario, de la siguiente manera: columns={'Columna 1':'Columna renombrada'}



Fuente: https://rfriend.tistory.com

Ordenar un dataframe



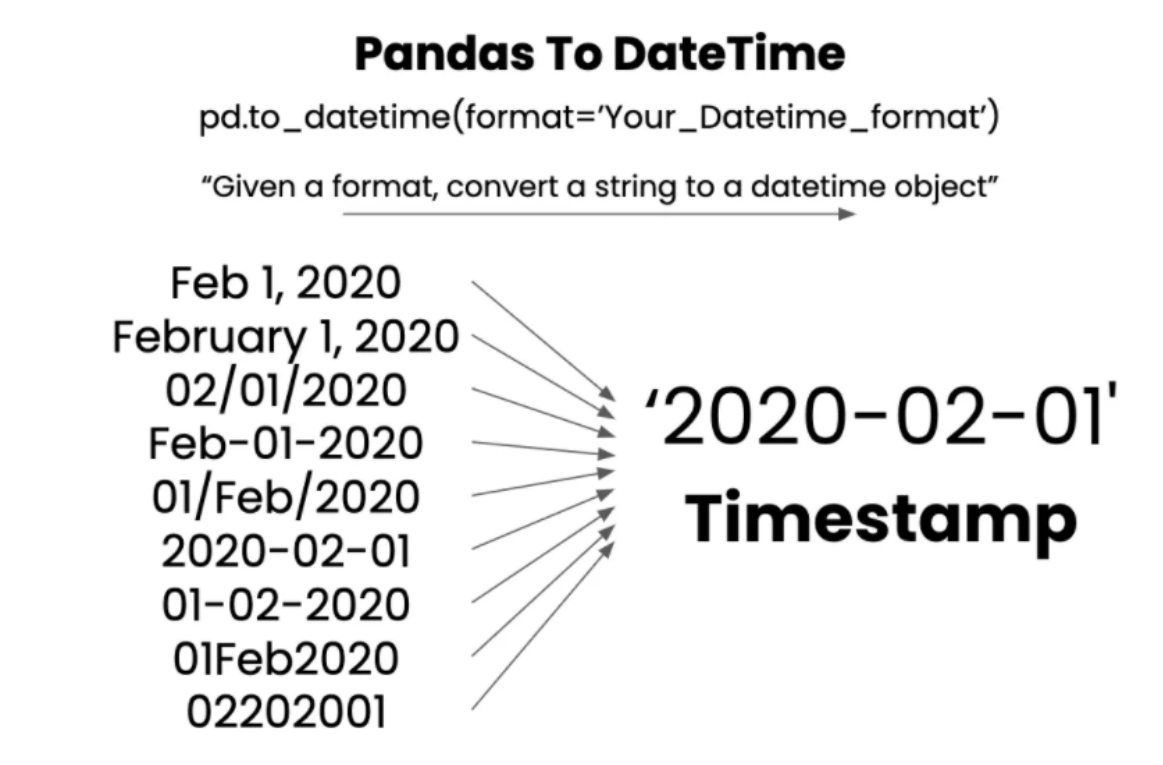
Fuente: https://datascienceparichay.com/article/pandas-sort-a-dataframe/

Para ordenar los valores y las filas respectivamente de un dataframe se pueden usar los siguientes métodos:

- df.sort_values(ascending=booleano) : Ordena los valores de manera ascendente o descendente según el argumento del parámetro. Si el parametro ascending es igual a True el orden es creciente y si es False decreciente.

- df.sort_index(ascending=booleano) : Devuelve el DataFrame que resulta de ordenar el respectivo índice. Al igual que en df.sort_values el argumento del parámetro ascending es True el orden es creciente y si es False decreciente.

Convertir una columna al tipo datetime



A menudo una columna contiene cadenas que representan fechas. Para convertir estas cadenas al tipo datetime se utiliza el siguiente método:

pd.to_datetime(columna, formato): Devuelve la serie que resulta de convertir las cadenas de la columna con el nombre columna en fechas del tipo datetime con el formado especificado en formato.

Aplicar funciones a columnas

Para aplicar funciones a todos los elementos de una columna se utiliza el siguiente método:

- df[columna].apply(f) : Devuelve una serie con los valores que resulta de aplicar la función f a los elementos de la columna con nombre columna del DataFrame df.

Añadir una fila a un DataFrame

- Para añadir una fila a un DataFrame se utiliza el siguiente método:

- df.append(serie, ignore_index=True) : Devuelve el DataFrame que resulta de añadir una fila al DataFrame df con los valores de la serie serie. Los nombres del índice de la serie deben corresponderse con los nombres de las columnas de df. Si no se pasa el parámetro ignore_index entonces debe pasarse el parámetro name a la serie, donde su argumento será el nombre de la nueva fila.

Eliminar filas de un DataFrame

Para eliminar filas de un DataFrame se utilizan el siguiente método:

df.drop(filas) : Devuelve el DataFrame que resulta de eliminar las filas con los nombres indicados en la lista filas del DataFrame df.

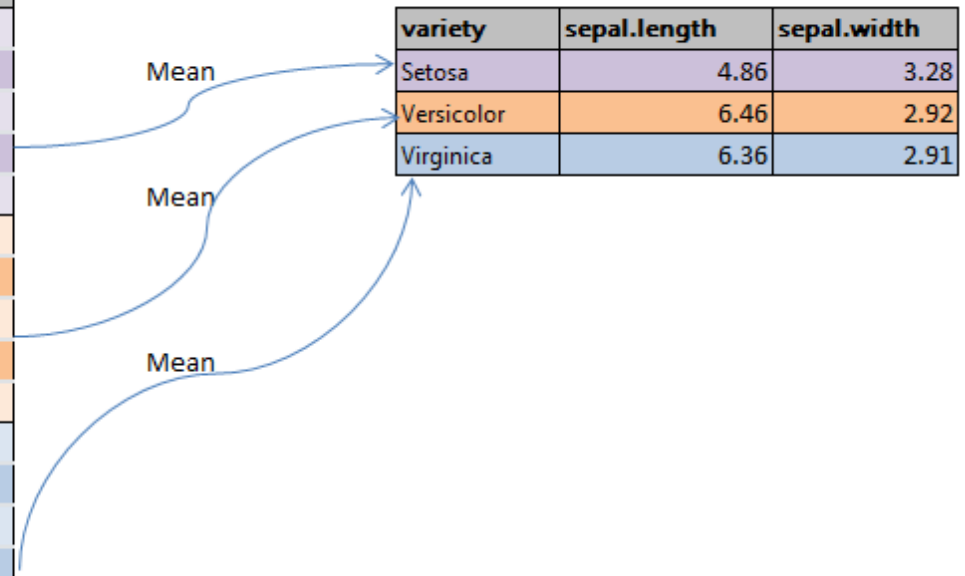
Eliminar las filas con datos desconocidos en un DataFrame

- Para eliminar las filas de un DataFrame que contienen datos desconocidos NaN o nulos None se utiliza el siguiente método:

- df.dropna(subset=columnas) : Devuelve el DataFrame que resulta de eliminar las filas que contienen algún dato desconocido o nulo en las columnas de la lista columna del DataFrame df. Si no se pasa un argumento al parámetro subset se aplica a todas las columnas del DataFrame.

Dividir un DataFrame en grupos

sepal.length	sepal.width	variety
5.1	3.5	Setosa
4.9	3	Setosa
4.7	3.2	Setosa
4.6	3.1	Setosa
5	3.6	Setosa
7	3.2	Versicolor
6.4	3.2	Versicolor
6.9	3.1	Versicolor
5.5	2.3	Versicolor
6.5	2.8	Versicolor
6.3	3.3	Virginica
5.8	2.7	Virginica
7.1	3	Virginica
6.3	2.9	Virginica
6.5	3	Virginica
7.6	3	Virginica
4.9	2.5	Virginica



variety	sepal.length	sepal.width
Setosa	4.86	3.28
Versicolor	6.46	2.92
Virginica	6.36	2.91

https://www.datasciencemadesimple.com/group-by-mean-in-pandas-dataframe-python-2/

Para dividir un DataFrame en grupos se utiliza el siguiente método:

- df.groupby(columnas).groups : Devuelve un diccionario con cuyas claves son las tuplas que resultan de todas las combinaciones de los valores de las columnas con nombres en la lista columnas, y valores las listas de los nombres de las filas que contienen esos valores en las correspondientes columnas del DataFrame df.

Convertir un DataFrame a formato ancho

Para convertir un DataFrame de formato largo a formato ancho (filas a columnas) se utiliza el siguiente método:

- df.pivot_table(index=filas, columns=columna, values=valores) : Devuelve el DataFrame que resulta de convertir el DataFrame df de formato largo a formato ancho. Se crean tantas columnas nuevas como valores distintos haya en la columna columna. Los nombres de estas nuevas columnas son los valores de la columna columna mientras que sus valores se toman de la columna valores. Los nombres del índice del nuevo DataFrame se toman de los valores de la columna filas.

Dataset

Name	Gender	Age
John	Male	45
Sammy	Female	6
Stephan	Male	4
Joe	Female	36
Emily	Female	12
Tom	Male	43

Pivot Table

Gender	%Gender	Age Group	Count
Male	50%	>18 years	2
		<18 years	1
Female	50%	>18 years	1
		<18 years	2

https://www.geeksforgeeks.org/how-to-include-percentage-in-pivot-table-in-pandas/